**Step 1: Video recording instructions**

1. Try to get the full body in the field of view (if using a smartphone, you may need to orient the phone vertically).
2. Try to capture at least one stride (i.e., two heel-strikes) with each leg.
3. Include an object with a known length in the field of view, and ensure that this object is placed at the same depth or distance from the camera as the person walking.
4. Try to minimize movements outside of the sagittal plane.
5. If using the Google Colab approach in Step 2, upload your videos to Google Drive.

**Step 2: OpenPose video analysis instructions (Google Colab version)**

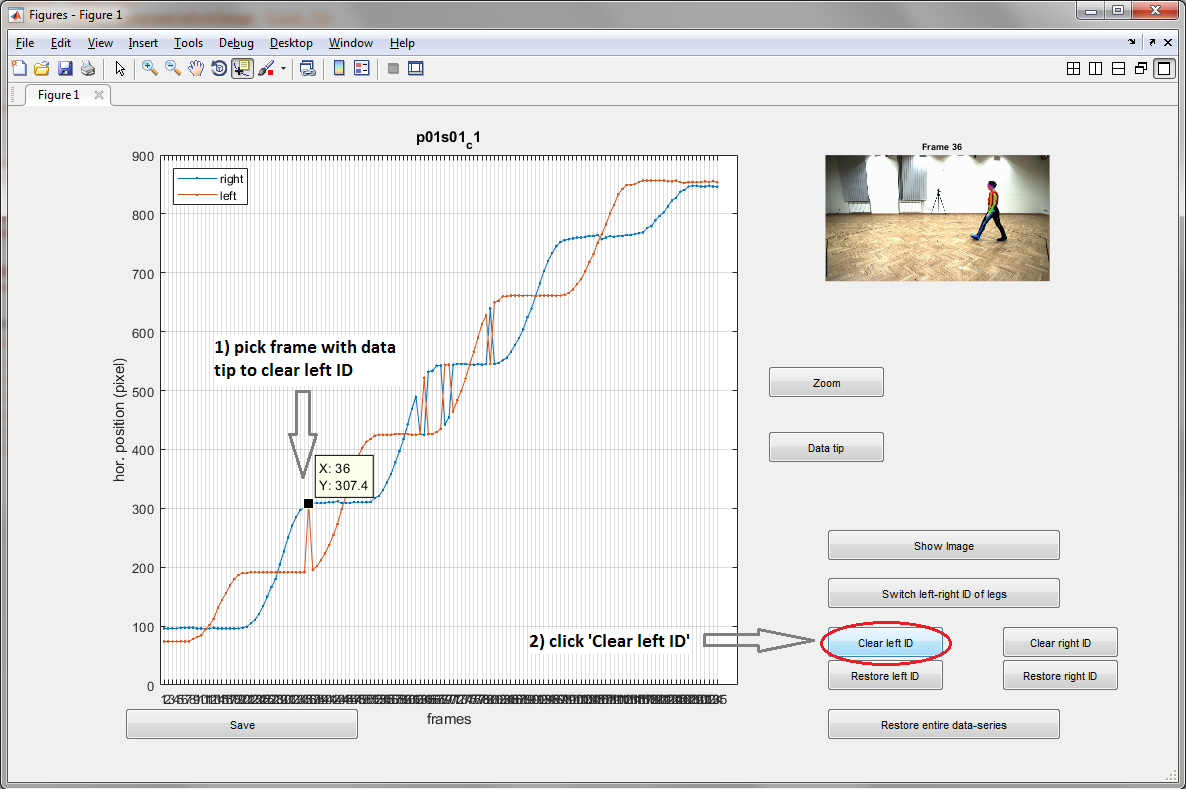
1. Go to colab.research.google.com.
2. Click Upload and upload the provided Google Colab notebook (OpenPose\_JS.ipynb).
3. Follow the step-by-step instructions within the Google Colab notebook.
4. Once the analysis is finished, download the labeled OpenPose output video and JSON output files for offline processing.

**Step 2 (*alternative to Google Colab, you do not need to do both*): OpenPose video analysis instructions (local version)**

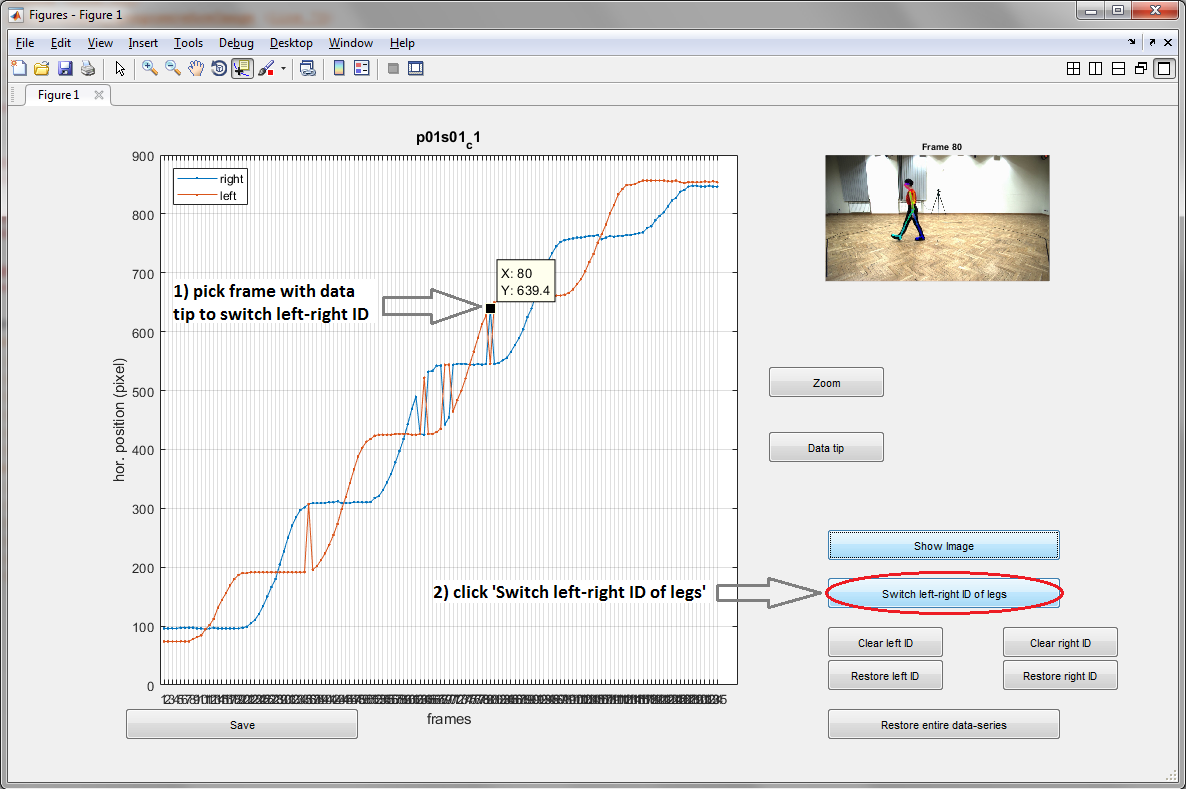
1. Go to https://github.com/CMU-Perceptual-Computing-Lab/openpose and follow instructions for installing OpenPose on your computer and running the OpenPose demo.

**Step 3: OpenPose video analysis instructions (MATLAB version)**

1. Place the 2 video files (original video and labeled OpenPose output video with keypoints shown) and JSON folder output from OpenPose (after extracting the JSON .rar or .zip file, if necessary) in the same folder.
2. Run *OpenPoseGaitAnalysis\_JS.m* in MATLAB (ensure that all functions that run below are included in the same folder as *OpenPoseGaitAnalysis.m*).
   1. *process\_openpose.m* runs first (*user input required*)
      1. Choose all JSON files in folder, then choose the original unlabeled input video, then choose the labeled OpenPose output video.
   2. *correctLegID\_openpose.m* runs second (*user input required*)
      1. A figure plotting the ankle marker trajectories (blue = right, orange = left) will now be plotted in the MATLAB Command Window.
      2. Use the zoom and data tips features in MATLAB to inspect and navigate ankle marker trajectories for each frame to:
         1. Clear specific data frames on the left or right leg trajectories. Click on the mislabeled data point on the trajectories plot and click one of the “Clear left ID” or “Clear right ID” buttons. This will clear keypoints: hip, knee, ankle, heel, big toe and small toe. See example below.



* + - 1. Switch specific data frames on the left or right leg trajectories. The legs are sometimes mistakenly switched for one another mid-trial; click on the mislabeled data point on the trajectories plot and click the “Switch left-right ID of legs” button. This will switch keypoints: hip, knee, ankle, heel, big toe and small toe. See example below.



* + 1. Use ‘Show Image’ to inspect OpenPose keypoints if needed.
    2. Use ‘Restore left ID’, ‘Restore right ID’ and ‘Restore entire data-series’ to revert to original data.
    3. Press ‘Save’ when finished.
  1. *gapFill\_filter\_openpose.m* runs third (*no user input required*)

1. No user input required.
   1. *findEvents\_openpose.m* runs fourth (*user input required*)
2. Plots excursions of ankle and MidHip for left and right legs. Positive peaks are heel-strikes and negative peaks are toe-offs. Use brush tool to delete spurious events.
3. Press ‘Save’ then ‘Close’ when finished.
   1. *extractScaling\_openpose.m* runs fifth (*user input required*)
4. Input known length (in meters) used for scaling.
5. Zoom in on points along walkway to be used to set scale (e.g., strips of tape in our manuscript). See example below.
6. Put data tip cursors at each point to set scale (2 data tips total; place one data tip first, then place the second data tip by either holding shift and clicking on the desired location, or right-click on the image, select ‘Create New Data Tip’, then place the second data tip).

A screenshot of a social media post

Description automatically generated

1. Press ‘Calculate Scaling’.
2. Press ‘Save’.
   1. *calculate\_gaitParameters\_jointAngles.m* runs sixth (*no user input required*)
3. No user input required.
   1. The ‘.mat’ file’ will now contain 6 structures: ‘data\_openpose’, ‘events\_openpose’, ‘frameInfo’, ‘videoInfo’, ‘gaitParameters’, and ‘jointAngles’ along with the name of the video analyzed (‘output\_name’).
4. ‘data\_openpose’ contains data matrices ‘raw\_data’ (raw input from JSON files), ‘corrected\_data’ (data with left-right ID errors corrected), ‘gapFill\_data’ (gap filled data) and ‘filt\_data’ (filtered data). Data matrices are organized as: time frames X keypoints (see list below) X coordinates (1, horizontal; 2, vertical). The struct ‘scaling’ contains the scaling factor used to dimensionalize pixel coordinates of OpenPose keypoints.
5. ‘events\_openpose’ contains the data frames of heel-strike and toe-off gait events.
6. ‘videoInfo’ contains MATLAB VideoReader variables used to display still frames from original video recording and labelled video recording.
7. ‘frameInfo’ contains logical vectors and matrices that indicate which frames have been corrected, switched and gap filled.
8. ‘gait parameters’ contains step times, stance times, swing times, double support times and step lengths for all steps and the average gait speed.

|  |  |  |
| --- | --- | --- |
| **OpenPose keypoints** | | (https:// github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/output.md) |
| 1 | Nose |
| 2 | Neck |
| 3 | Right should |
| 4 | Right elbow |
| 5 | Right wrist |
| 6 | Left shoulder |
| 7 | Lelft elbow |
| 8 | Left wrist |
| 9 | Mid hip |
| 10 | Right hip |
| 11 | Right knee |
| 12 | Right ankle |
| 13 | Left hip |
| 14 | Left knee |
| 15 | Left ankle |
| 16 | Right eye |
| 17 | Left eye |
| 18 | Right ear |
| 19 | Left ear |
| 20 | Left big toe |
| 21 | Left small toe |
| 22 | Left heel |
| 23 | Right big toe |
| 24 | Right small toe |
| 25 | Right heel |

1. ‘jointAngles’ contains sagittal plane hip, knee and ankle angles.